Eye-To-Hand Coordination: A Coarse-Calibrated Approach

Shahram Jafari **Ray Jarvis**

Perceptive and Intelligent Machines in Complex Environments (PIMCE)
Monash University
VIC 3800, Australia

<Shahram.Jafari, Ray.Jarvis>@eng.monash.edu.au

Abstract

This paper integrates different novel concepts to perform eye-to-hand coordination and manipulation to realise a working robot named COERSU. One of the salient features of our system is that it is not highly dependent to the camera-robot calibrations. Our coarse-calibrated eye-to-hand visual servoing uses monocular vision and primitive tactile information to extract 3D information of the scene. A robust genetic tuner is briefly presented to optimise the parameters of the image segmentation and edge detection. Some simple methods for scene classification, pose detection and final verification were also implemented to perform intelligent grasping. Experimental results from COERSU in a table-top scenario to manipulate some soft objects (fruit and egg) are also provided to validate these methods (snapshots and videoclip).

1 Introduction

The process of visual understanding begins with the image data, its segmentation into coherent components, the recognition and pose of them and potentially an action plan related to these processes. Meanwhile, the goal of image segmentation is to partition an image into meaningful homogeneous regions [Le Moigne, 1992; Hofman, 2000; Gong and Yang, 2001]. The main tasks commonly involved in a robotic eye-to-hand (as distinct from eye-in-hand) coordination for object grasping can be summarized as follows: First, the target and its pose are perceived through sensors. Then, using the necessary transformation, the pose of the target is mapped to the 3D robot frame, and finally, using the kinematics of the robot arm, a proper trajectory is planned to grasp the object. Since we want our robot to perform some human-centric tasks, we have avoided using laser or sonar sensors to detect the range of the objects.

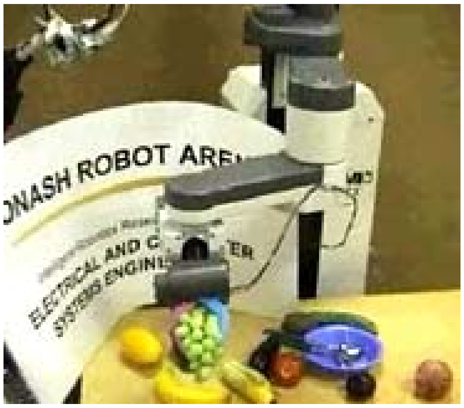

Figure 1. COERSU and a table-top scenario

Study of the cooperation between a camera and a robotic manipulator started some time ago [Castore and Crawford, 1984; Kabuka, Desoto et al., 1988; Rao, Medioni et al., 1988] since then, there have been numerous research projects conducted to study fast and reliable robotic hand-eye coordination. Most of these focus on the estimation of a feature Jacobian matrix [Hosoda and Asada, 1994; Hutchinson, Hager et al., 1996] mapping the motion of the robot arm to the changes in the image frame. These approaches concentrate on robots with cameras on their hands [Hashimoto, Kimoto et al., 1991; Malis, 2004]. Since human-like robots do not have eye in hand, we have avoided this approach. Another reason is that we wanted to observe the whole arm in the context of the obstacles in the scene so as to easily avoid them.

Image-based visual servoing using stereo vision (binocular camera) has been reported in the literature [Hosoda, Sakamoto et al., 1995; Cervera, Berry et al., 2002]. Although a grey scale stereo vision system was part of our platform, the accurate range measurements of the tactile sensors (at the end of the tooltip) outperformed the noisy results [Konolige and Beymer, 2002] of the stereo vision. In relative visual servoing, the robot arm could be easily manipulated to avoid object collision. This makes planning the trajectory easier in this approach. In our set-up, first the robot tooltip goes directly above the target and then moves downwards to the target to grasp and pick it up. This is a reasonably good trajectory

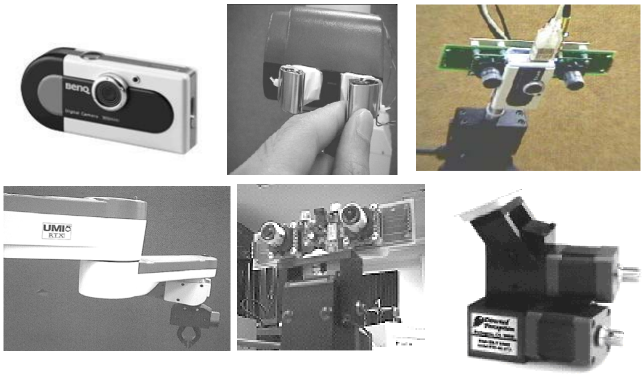

Figure 2. Basic hardware components (row-wise from left): Color camera, Primitive tactile sensors, Head of COERSU, Arm of COERSU, Grey scale stereo vision system, Pan-tilt unit (neck).

planning strategy in the context of obstacle avoidance in table-top scenarios.

We first present our hardware platform in Section 2. Then, different implemented methods are discussed in Section 3. Finally snapshots and video-clip are shown at the end to demonstrate the effectiveness of our methodologies.

2 Hardware Description

In order to validate our methods, we set up COERSU to pick and place some objects on a dining table (Fig. 1). The major hardware components of COERSU are (Fig. 2): 1. an anti-speckle USB color camera of resolution 640*480 acting as a visual sensor, 2. a gray-scale stereo camera pair producing left and right gray-scale interlaced images and a rough estimation of range is obtained by the disparity image [Konolige and Beymer, 2002]. 3. an RTX UMI robotic manipulator acting as an arm for COERSU, 4. a set of primitive tactile sensors around the tooltip of the robot and their relevant interface circuitry, 5. a pan-tilt unit resembling a neck for COERSU. 6. a frame-grabber which writes left and right frames given by the stereo camera pair directly to the disk. Therefore, COERSU can easily move its eye to adapt to movements of the arm and different locations in the scene (Fig. 1). The noisy result of grey scale stereo vision system [Konolige and Beymer, 2002] was outperformed by the accurate measurement of depth using tactile sensors.

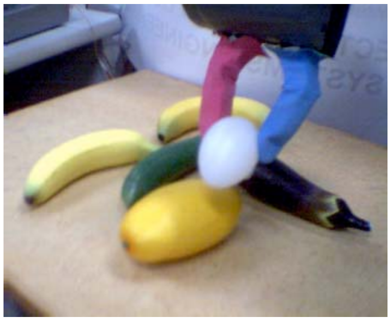

Figure 3. An egg is grasped and picked up by COERSU in a table-top scenario.

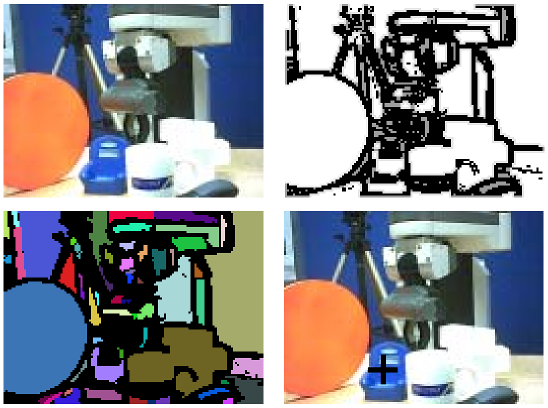

Figure 4. Result of online object recognition based on the genetically tuned parameters row-wise from left: Original image frame, Result of edge detection, Tuned segmentation, Object recognition (obj. shown by cross).

2.1 Tactile Sensors

Some of the advantages in our implementation of tactile sensors can be summarized as: 1.Grasping soft objects (Fig. 3) such as fruit on a dining table without squeezing them. 2. Grasping free-size objects autonomously. 3. Assisting the vision system in obstacle avoidance by verifying the existence of fixed objects (estimating table depth by touching it Section 3.3). 4. Touching and verifying in the last stage before final grasping (Section 3.7).

3. Methodologies

3.1 Genetic off-line tuner

Some pre-processing of the image frames are necessary before applying the genetic tuner. This pre-processing consists of, but is not limited to, averaging (zoom out), registration and normalisation. A genetic tuner [Jafari and Jarvis, 2003] is applied in order to optimise the parameters of the image processing in the methods considered here. Although using this tuner makes the system more complicated, it is preferable to apply it to obtain a higher degree of precision.

The genetic tuner makes the individual processes such as detecting the targets and the edges, segmenting the image, finding the tooltip in the image, and measuring the centroid and other features of objects more robust as shown in Fig. 4. One of the achieved goals is to speed up the process of real-time segmentation by eliminating any tuning (optimisation) sessions from the on-line process

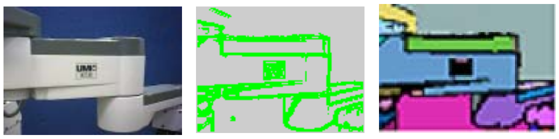

Figure 5. Representative ground truth image (from left to right): The original image, Result of Sobel edge detector, Result of region-edge segmentation

and carrying them out off-line. Nine parameters of the visual perception that are mainly related to the region-edge segmentation are tuned using the genetic algorithm. A novel idea for automatic evaluation of the chromosomes has been applied by comparing a group of ground truth labelled images (Fig. 5) with the output of the system based on the candidate chromosome.

The comparison is done through counting the number of mismatching edge-pixels in the output of the segmentation and the ground truth images. The chromosome fitness function is calculated based on this error using the following equation:

$$Fitness_{chromosome\_i} = \frac{1}{error_i + 1} \qquad (1)$$

$Fitness_{chromosome\_i}$: Fitness evaluated for chromosome i,

$error_i$: Error caused by applying that chromosome.

Based on a trial and error, we chose a population size of 20 chromosomes for each generation and a maximum number of 30 generations for evolution. The tuning process takes about one hour for 10 different ground truth images. The genetic function cross-over (with a probability of 0.5) was a random weighted averaging of the parent chromosome genes where the weights were the fitness functions of the parent chromosomes. Thus, we used a mix instead of the conventional cross-over to generate off-spring and more favourable results were achieved using this approach.

Reproduction was based on a roulette-wheel selection of the parent. In order to provide occasional disturbances in the process of reproduction or cross-over, a single gene in the offspring chromosome is randomly selected and mutated with the probability 0.1. In addition, in order not to miss the chromosome that has the maximum fitness value, the elite chromosome is directly transferred to the offspring generation.

3.2 Scene analysis

The major goal of scene analysis is to be able to determine what and where the objects are in the scene, and the possible interrelationship between the objects in order to manipulate them by a robot. A simple k-nearest neighbourhood method was considered because the objects of our concern were fairly dissimilar. We decided to apply five-nearest neighbourhood classification to perform scene analysis and the results were sufficiently accurate to detect a particular target with an overall success rate of 9 out of 10 (90%).

This process was performed prior to the manipulation of the target object so that the robot arm could not influence the classification of the objects. We trained our system with fifty different known object-pose cases. This consists of five samples for each of the ten prototype objects. These objects were some fruit such as, banana, cucumber, kiwi, eggplant, capsicum and some other kitchen utensil: mug, plate, spoon, fork, and knife. Objects were chosen based on their simplicity for manipulation by a robot hand with two fingers. For each of the objects we considered five different poses on the table in order to extract enough feature variations.

Classification features

The best combination of features will produce the greatest difference in the feature values of significantly different shapes and the least difference for similar shapes. The features used in our classification methods were chosen based on trial and error and consisted of, Perimeter, Area, Shape features A and B described below[Hu, 1962] and Colour components (Red, Green, Blue).

The following two moment invariants are computed to achieve two goals: primarily to get some shape descriptors, and secondly, to obtain scale, translation, and rotation invariant features for the objects of interest.

$$A = \mu_{20} + \mu_{02}, \qquad (2)$$

$$B = \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \qquad (3)$$

Where the central moments μ_{20}, μ_{11}, μ_{02} used above (normalized with respect to size) are obtained from the ordinary moments:

$$\mu_{pq} = {v_{pq}} \Big/ {v_{00}^{(p+q+2)/2}}$$

$$v_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(x-\bar{x})^p(y-\bar{y})^q f(x,y) \qquad (4)$$

where $\bar{x} = {m_{10}} / {m_{00}}$, $\bar{y} = {m_{01}} / {m_{00}}$,

$$m_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} x^p y^q f(x,y) \qquad (5)$$

M,N: boundaries of the object rectangle. $f(x,y)$: binary representation of the object in the segmented image.

In order to have a measure of similarity between an unknown object and the prototypes the weighted Euclidean distance metric was used [Jafari and Jarvis, 2003]. Since the Euclidean distance function treats every dimension equally, it was necessary to normalize the data. By analyzing all of the training data in a pre-processing stage, we determined the range of every attribute and transformed the entire dataset appropriately, Fig. 6.

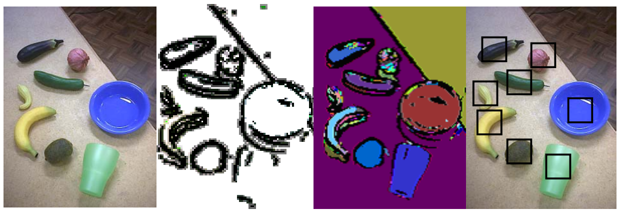

Figure 6. Result of online object recognition based on the genetically tuned parameters (from left to right): (a) original image frame (b) result of edge detection (c) tuned segmentation (d) object classification.

3.3 Off-line rough estimation of the robot-image transformation

The rough estimate of the transformation is obtained using the following procedures:

1. The camera's pose with respect to the robot's 3D coordinate frame is roughly measured.

2. By using the tactile sensor and touching the surface of the table, we find the coordinates of an arbitrary point on the table with respect to the robot 3D frame, Fig. 7.

It is also used to verify the noisy result of the disparity images [Konolige and Beymer, 2002] and therefore we get a more accurate range information. Calibration of the stereo-vision parameters can be also carried out by using this information; however, we did not use the stereo measurements during our visual-servoing and the results were satisfactory.

Then, the 3D point-to-point distance between that point and the camera origin (from step 1) is obtained using (6).

$$D_{(C)\leftrightarrow(M)} \cong \sqrt{\sum_{k=1}^{3}(C_k - M_k)^2} \qquad (6)$$

where $D_{(C)\leftrightarrow(M)}$ is the distance between the camera C and the point M and k represents x,y or z coordinate.

The value obtained in Step 2 is used as an estimate for the range of the target from camera [Schilling, 1990]. Since we assume the table as a fixation that the objects are located on top of and the targets are within the robot's workspace, this is a good estimate.

3. We construct an un-calibrated perspective transformation matrix [Schilling, 1990] based on the values obtained in the previous steps.

Although the final transformation matrix is not accurate, it is accurate enough for the purpose of alignment discussed next.

3.4 Method used to align the tooltip with the target

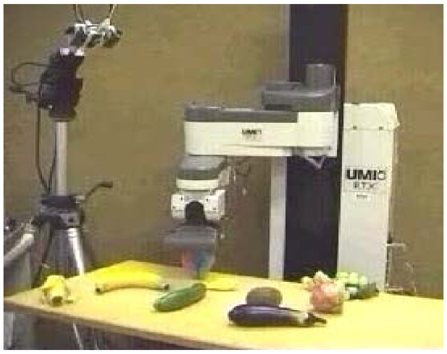

Figure 7. Table as a fixature that the objects are located on top of and the targets are within the robot's workspace (off-line table depth estimation).

Here, by alignment we mean placement of the centroid of the tool gripper in line with the centroid of the target object with respect to the camera point of view (i.e. not directly above the target).

The procedure for the alignment

Step 1: The centroid of the target position is recognized autonomously in the image frame by object recognition (Section 3.2).

Step 2: Using the un-calibrated transformation obtained in section 3.3, we find *x* and *y* coordinate of the target with respect to the robot coordinate frame.

Step 3: Using the inverse kinematics, we instruct the robot to move the tooltip to the *x* and *y* coordinate of the target found in Step 2. This movement is along a *z*-plane (height) fairly far from the table.

Step 4: In the image frame, the difference between the new position of the centroid of the tooltip and position of the target is obtained in both row and column directions (Tooltip of the robot is detected based on the robot to image transformation and the color of the two fingers).

Step 5: This difference is used to find the next adjustment of the tooltip. The last two steps of the procedure are carried out until the difference is less than a threshold value.

3.5 Relative visual servoing

Biologically-inspired visual servoing is currently receiving more focus in the literature [Kragic and Christensen, 2003]. A human-inspired approach for locating the target using a monocular camera can be achieved as follows: First, a few planned movements of the hand are made. Then the relative information between the target pose and the hand, picked up by the eye, and the estimation about the hand movements are used to judge the position of the target. We could emulate this behavior in three major steps: initially, lining up the centroid of the target with the center of the robot tooltip; secondly, moving downwards maintaining this alignment; and finally grasping the object when the tooltip senses the target. This approach has the following disadvantages:

- This approach is time-consuming because of the necessity to maintain the condition of alignment, which in turn, requires a continuous visual feedback.
- If we have other restrictions on the trajectory of the tooltip, for instance, to avoid obstacles, the above approach is not suitable. It is possible that the wrist of the robot and/or the tooltip could disturb the objects surrounding the target (due to the structure or bulkiness of the wrist of the robot such as the RTX UMI robot arm).
- Considering the target, this approach can disturb a vertically positioned target by knocking it down.

Therefore, in our approach we modified this behavior by taking just two points of alignment (Section 3.4)[Jafari,

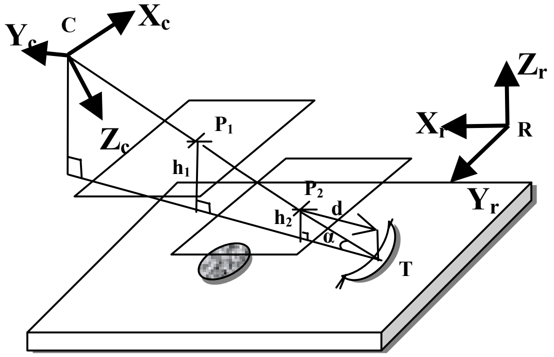

Figure 8. Relative Visual Servoing

R, Robot coordinate frame; C, camera coordinate frame; P_1, P_2, two alignment points; h_1, h_2, their heights from the table; d, horizontal motion adjustment; α, view point angle

Jarvis et al., 2004] and then using this information to find the horizontal motion required to place the tooltip directly above the target based on the trigonometric relations. It is shown that two points of alignment contain sufficient information about the depth of an object on the table. Then, move downwards to pick it up as shown in Fig. 8.

3.6 Intelligent grasping

Pose estimation of the target is necessary for a reliable manipulation especially when the object is elongated such as a cucumber or a banana and the tooltip of the robot is a two finger grasping tool as in COERSU. It is difficult to grasp an elongated object on the table if we have no information about the yaw (horizontal angle) of that object. Orientation detection of the objects was carried out based on the calculation of their principal and minimal axes using central moments [Schilling, 1990]. Then, during the grasping phase, the fingers were placed parallel to the principal axis of the object and perpendicular to the minimal axis based on a linear estimation function. The error in pose estimation was less than 2 degrees, which was quite satisfactory for our grasping experiments.

We applied a central moment technique [Schilling, 1990; Mikolajczyk and Schmid, 2003; Tahri and Chaumette, 2003] to derive the slope, θ of the principal axis:

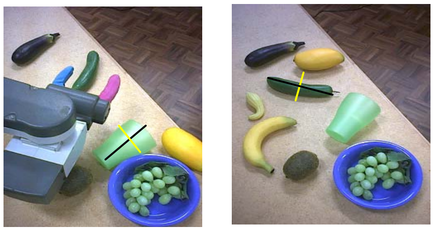

Figure 9. Typical results of online object pose estimation (left) mug, (right) cucumber.

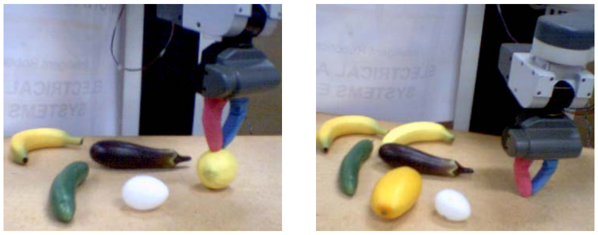

Figure 10. COERSU verifies the presence of a lemon

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{\mu_{20}-\mu_{02}} \tag{7}$$

Where the central moments μ_{20}, μ_{11}, μ_{02} used above (normalized with respect to size) are obtained from (4,5). After calculating the principal axis, the minimum axis is considered as the line perpendicular to it. The result of pose detection for two sample objects is shown in Fig. 9.

3.7 Verification stage

Provided that the target sufficiently stimulates the tactile sensors, (i.e. not hollow inside like a mug or liquid) we can consider the last stage of moving downwards to make sure that the target is located underneath. This is to some extent similar to a blind human trying to locate a particular object on the table. The process can be summarized as follows: in order to make sure the target will be in between the fingers of the tool-gripper, COERSU performs a vertical touch down with its wrist in a position that the tooltip tactile sensors point downwards. Then, the tactile sensors are stimulated after touching a surface and the arm stops moving further (Fig. 10). The difference between the previously measured depth of the table and the current depth estimation is a verification criterion:

If the difference is less than a threshold value, it means that COERSU has touched the table instead of the target and the process of visual servoing has to be repeated. However, if the difference is greater, COERSU starts the next stage of grasping the target.

4 Conclusion

We tried to cover different aspects of a coarse-calibrated hand-eye coordination for a working platform named COERSU. Our journey started with an “evolved” image scene interpretation using genetic algorithm. Specifically, the segmentation method was based on the genetically optimized [Jafari and Jarvis, 2003] parameters of the scene, then we briefly covered a human-inspired method, named Relative Visual-Servoing, for autonomously picking and placing a target object in a table-top scenario (Fig. 11 and video-clip). Some simple methods for object classification and pose detection were also implemented for intelligent grasping.

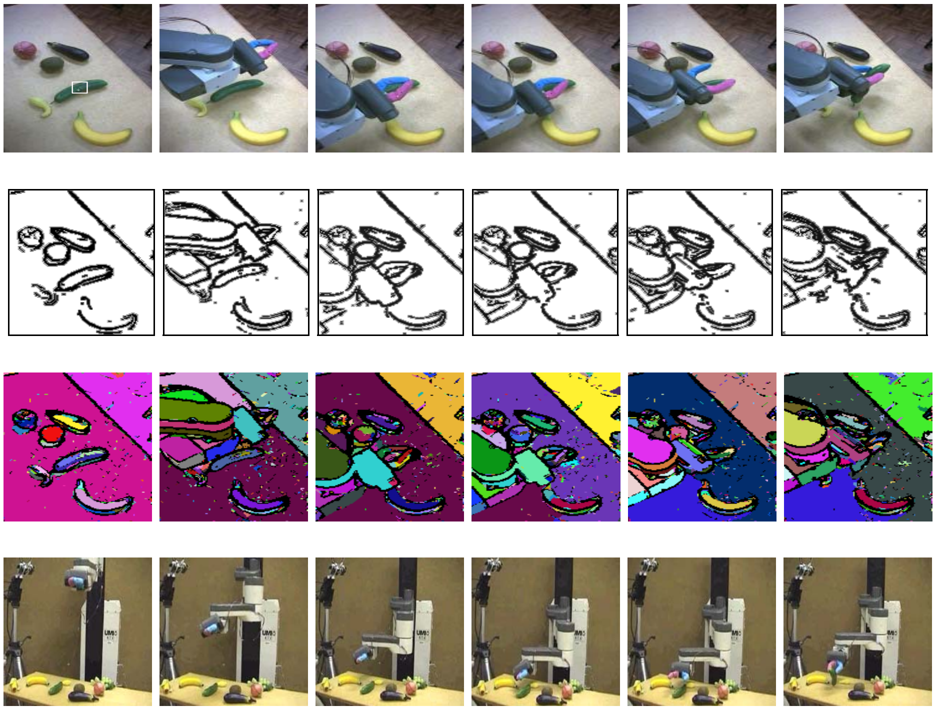

Figure 11. Relative visual servoing in response to the command: 'pick up the cucumber' (column-wise from left) a) Frame No.1-initial scene b) Frame No.2-first positioning based on uncalibrated transformation c) frame No.13-first alignment d) frame No.19-second alignment e) Frame No.20-final adjustment f) Frame No.21-grasping the cucumber. (row-wise from top) i) original image ii) result of edge detection iii) segmentation iv) video sequence taken by a camcorder

References

[Castore, G. and C. Crawford, 1984]. *From solid model to robot vision*. Robotics and Automation. Proceedings. 1984 IEEE International Conference on.

[Cervera, E., F. Berry, et al., 2002]. *Is 3D useful in stereo visual control?* IEEE Int. Conf. on Robotics and Automation.

[Gong, M. and Y. H. Yang, 2001]. *Genetic-Based Multiresolution Color Image Segmentation*. Vision Interface 2001, Ottawa, Ontario.

[Hashimoto, K., T. Kimoto, et al., 1991]. *Manipulator control with image-based visual servo*. IEEE International Conference on Robotics and Automation.

[Hofman, I. D., 2000]. Three dimensional scene analysis using multiple view range data. *Intelligent Robotics Research Centre (IRRC)*. Clayton, Monash University, Australia.

[Hosoda, K. and M. Asada, 1994]. *Versatile visual servoing without knowledge of true Jacobian*. IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems.

[Hosoda, K., K. Sakamoto, et al., 1995]. *Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3D reconstruction*. IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems.

[Hu, M.-K., 1962]. "Visual pattern recognition by moment invariants." *Information Theory, IEEE Transactions on* **8**(2): 179-187.

[Hutchinson, S., G. D. Hager, et al., 1996]. "A tutorial on visual servo control." *Robotics and Automation, IEEE Transactions on* **12**(5): 651-670.

[Jafari, S. and R. A. Jarvis, 2003]. *A Genetic Off-line Tuner for Robotic Humanoid Visual Perception*. IEEE International Congress on Evolutionary Computation (CEC-2003).

[Jafari, S., R. A. Jarvis, et al., 2004]. *Relative Visual Servoing*. IEEE Conference on Robotics, Automation and Mechatronics (RAM), Singapore.

[Kabuka, M., J. Desoto, et al., 1988]. "Robot vision tracking system." *Industrial Electronics, IEEE Transactions on* **35**(1): 40-51.

[Konolige, K. and D. Beymer, 2002]. SRI Small Vision System User's Manual, Software version 2.2f**:** 67.

[Kragic, D. and H. I. Christensen, 2003]. *Biologically motivated visual servoing and grasping for real world tasks*. IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2003).

[Le Moigne, J., 1992]. *Refining Image Segmentation by Integration of Edge and Region Data VO - 2*. Geoscience and Remote Sensing Symposium, 1992. IGARSS '92. International.

[Malis, E., 2004]. "Visual Servoing Invariant to Changes in Camera-Intrinsic Parameters." *Robotics and Automation, IEEE Transactions on* **20**(1): 72-81.

[Mikolajczyk, K. and C. Schmid, 2003]. *A performance evaluation of local descriptors*. Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on.

[Rao, K., G. Medioni, et al., 1988]. *Robot hand-eye coordination: shape description and grasping*. Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on.

[Schilling, R. J., 1990]. *Fundamentals of robotics: analysis and control*, Prentice-Hall Inc.

[Tahri, O. and F. Chaumette, 2003]. *Application of moment invariants to visual servoing*. Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on.